REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1-23 are pending in the application. The Examiner additionally stated that claims 1-23 are rejected. By this amendment, claims 1, 3, and 5 have been amended. Hence, claims 1-23 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

In the Specification

The Examiner objected to the specification because the abstract is greater than 150 words. Correction was required. By this amendment, Applicant has amended the abstract such that it is has 150 words or less. In addition, Applicant has amended the specification to secure a substantial correspondence between the claims amended herein and the remainder of the specification. No new matter is presented.

In the Claims

Rejections Under 35 U.S.C. §112

The Examiner rejected claims 1-23 under 35 U.S.C. 112, first paragraph, as failing to comply with a written description requirement. In particular, the Examiner noted that the claims contain subject matter which was not described in the specification in such as was as to reasonably convey to one skilled in the relevant art that the inventors, at the time the application was filed, had possession of the claimed invention. The Examiner further noted that Applicant indicates that the bypass structure is within the data cache, by that it is not clear where this is taught in the specification. The Examiner pointed out that the specification teaches that the bypass structure is coupled to the data cache, but is not within the data cache.

By this amendment, Applicant has amended independent claims 1, 3, and 5 to indicate that the bypass structure is coupled to the data cache. Accordingly, it is respectfully requested that the rejections of claims 1-23 be withdrawn.

Rejections Under 35 U.S.C. §103(a)

The Examiner rejected claims 1-6, 10, 12-13, and 22-23 under 35 U.S.C. 102(a) as being unpatentable over Dubey et al., U.S. Patent No. 5,724,565 (hereinafter, Dubey) in view of Meier, U.S. Patent No. 6,523,109 (hereinafter, Meier). Applicant respectfully traverses the Examiner's rejections.

Prior to providing a claim-by-claim analysis, a brief overview of the teachings of both Dubey and Meier are provided below to aid the Examiner during reconsideration of the claims as amended herein.

Dubey teaches and method and system for processing instruction threads. Execution is initiated by a processing system of a first set of instructions including a particular instruction. The particular instruction includes an indication of a second set of instructions. In response to execution of the particular instruction and to the processing system being of the first type, the processing system continues executing the first set while initiating execution of the second set. In response to execution of the particular instruction and to the processing system being of a second type, the processing system continues executing the first set without initiating execution of the second set (Abstract). The apparatus includes an instruction cache having multiple ports 115-1, 115-2, etc., that enable simultaneous porting of instructions to instruction threads being executed in parallel. The apparatus also has a bank of program counters 120-1, 120-2, etc., that each tracks the execution of a certain thread. The apparatus further has bank of dispatchers 140-1, 140-2, etc., where each dispatcher is associated with a specific program counter 120-1, 120-2, etc., and is capable of receiving instructions from one instruction cache port 115-1, 115-2, etc (col. 6, line 65 - col. 7, line 60). In an alternative embodiment, Dubey teaches a store queue for speculative stores, where the processor coordinates memory dependencies between concurrently executed speculative and non-speculative code paths. If the processor encounters a store operation in the non-speculative path and a load operation in the speculative path, then the processor resolves the two operations so that the load operation involves the correct information (col. 28, lines 19-26).

Meier teaches a processor that includes a store queue configured to detect a hit on a store queue entry for a load being executed, and to forward data from the store queue entry to provide a result for the load (Abstract). Meier's invention is taught and contemplated in the context of a single-thread, super-scalar processor (Fig. 1; col. 3, line 45 - col. 7, line 56).

Applicant's invention, on the other hand, is directed toward a multi-streaming microprocessor core that executes instruction streams running within the multi-streaming microprocessor core at any time. The core includes instruction queues, a bypass structure, and address matching logic. Each of the instruction queues correspond to each of the instruction streams, and each include: a read pointer, for pointing to an oldest instruction, said oldest instruction having not yet been dispatched; a write pointer, for pointing to a newest valid instruction; first instructions, for dispatch to one or more functional units; store instructions, for dispatch to a data cache, wherein said store instructions direct write operations; and load instructions, for dispatch to said data cache, wherein said load instructions direct read operations. In addition, the each of the instruction queues retains up to 8 instructions already dispatched so that they can be dispatched again in the case that a short backward branch is encountered. The bypass structure is coupled to the data cache and receives the store instructions. The bypass structure has multiple elements, where, if the write operations hit in the data cache, thenb data corresponding to the write operations are stored in one or more of the elements in the bypass structure before the data is written to the data cache. The address matching logic is coupled to the bypass structure, and receives the load instructions, where the read operations use the address matching logic to search the elements of the bypass structure to identify and use any one or more of the elements representing more recent data than that stored in the data cache.

With specific reference to claims 1, 3, and 5, the Examiner noted that Dubey teaches a central processing unit for processing multiple parallel instruction threads, where the CPU includes a plurality of instruction buffers which correspond to the multiple instruction threads. The Examiner further noted that the CPU includes one or more functional units and a data cache. The Examiner stated that Dubey teaches different types of instructions, including branch instructions, load instructions, and store instructions. And the Examiner noted that Dubey also suggests a store queue and resolving memory dependencies between loads and stores.

It was noted that Dubey does not teach address matching logic and switching logic in association with the store queue, but that Meier teaches a store queue for a data cache. Meier's store queue includes a plurality of entries. The Examiner also stated that a comparison between a load address and entries in the store buffer is performed by a store queue number assignment circuit. The Examiner also noted that Meier teaches a merge/align circuit in the data cache for merging bytes from the store queue with bytes from the data cache, and that Meier also teaches that a load operation may match on multiple elements in the store queue. The Examiner concluded that it would have been obvious to one of ordinary skill in the art to have modified the store queue of Dubey to include the address matching and switching logic suggested by Meier, because Meier teaches that such a store buffer implementation would conserve the amount of circuitry used and decreases average load latency, as well as optimizing code sequences.

Claim 1 as amended herein is provided below for each of reference.

1. A multi-streaming microprocessor core, for executing instruction streams running within the multi-streaming microprocessor core at any time, the multi-streaming microprocessor core comprising:

instruction queues, each corresponding to each of the instruction streams, said each of said instruction queues comprising:

a read pointer, for pointing to an oldest instruction, said oldest instruction having not yet been dispatched;

a write pointer, for pointing to a newest valid instruction;

first instructions, for dispatch to one or more functional units;

store instructions, for dispatch to a data cache, wherein said store instructions direct write operations; and load instructions, for dispatch to said data cache, wherein said load instructions direct read operations;

wherein said each of said instruction queues retains up to 8 instructions already dispatched so that they can be dispatched again in the case that a short backward branch is encountered;

a bypass structure, coupled to said data cache, for receiving said store instructions, said bypass structure comprising multiple elements, wherein, if said write operations hit in said data cache, data corresponding to said write operations are stored in one or more of said elements in said bypass structure before said data is written to said data cache; and

address matching logic, coupled to said bypass structure, for receiving said load instructions, wherein said read operations use said address matching logic to search said elements of said bypass structure to identify and use any one or more of said elements representing more recent data than that stored in said data cache.

As alluded to above in the brief overview, the multi-streaming microprocessor core as recited in various embodiments provided by claims 1, 3, and 5, includes, in combination with other elements, instruction queues that each correspond to each of the instruction streams. Each of the instruction queues has: a read pointer, for pointing to an oldest instruction not yet been dispatched; a write pointer, for pointing to a newest valid instruction; first instructions, for dispatch to one or more functional units; store instructions, for dispatch to a data cache, wherein said store instructions direct write operations; and load instructions, for dispatch to said data cache, wherein said load instructions direct read operations. In addition, the each of the instruction queues retains up to 8 instructions already dispatched so that they can be dispatched again in the case that a short backward branch is encountered. The multi-streaming microprocessor core also includes a bypass structure that is coupled to the data cache and that receives the store instructions. The bypass structure has multiple elements, where, if the write operations hit in the data cache, then data corresponding to the write operations are stored in one or more of the elements in the bypass structure before the data is written to the data cache. The address matching logic is coupled to the bypass structure, and receives the load instructions, where the read operations use the address matching logic to search the elements of the bypass structure to identify and use any one or more of the elements representing more recent data than that stored in the data cache.

Applicant respectfully disagrees with the Examiner's rejections of claims 1, 3, and 5 and to his characterization of the teachings of both Dubey and Meier. First, as noted above, Dubey teaches an apparatus that includes an instruction cache having multiple ports 115-1, 115-2, etc., that enable simultaneous porting of instructions to instruction threads being executed in parallel. The apparatus also has a bank of program counters 120-1, 120-2, etc., that each tracks the execution of a certain thread. Applicant's invention, in contrast, has instruction queues that each correspond to each of the instruction streams. Each of the instruction queues has: a read pointer, for pointing to an oldest instruction not yet been dispatched; a write pointer, for pointing to a newest valid instruction; first instructions, for dispatch to one or more functional units; store instructions, for dispatch to a data cache, wherein said store instructions direct write operations; and load instructions, for dispatch to said data cache, wherein said load instructions direct read operations. In addition, the each of the instruction queues retains up to 8 instructions already dispatched so that they can be dispatched again in the case that a short backward branch is encountered. Applicant has searched the cited reference and finds that Dubey utterly fails to teach, allude to, hint, or even suggest instruction queues that have a read pointer, for pointing to an oldest instruction not yet been dispatched and a write pointer, for pointing to a newest valid instruction. Furthermore, Dubey does not provide any motivation whatsoever that would lead one skilled in the art to provide for instruction queues that retain up to 8 instructions already dispatched so that they can be dispatched again in the case that a short backward branch is encountered. This is because Dubey is addressing the problem initiating parallel execution of a second set of instruction responsive to execution of a particular instruction in a first thread that indicates the second set of instructions. Since Dubey fails to teach the two pointers and the instruction retention limitations, it also follows that he fails to teach such limitations where configured instruction queues provide their instructions to a data cache coupled to a bypass structure and address matching logic that are configured to receive instructions from the instruction queues.

As noted above, Meier teaches a single-thread processor that includes a store queue configured to detect a hit on a store queue entry for a load being executed, and to forward data from the store queue entry to provide a result for the load. Applicant has searched the teachings of Meier, and is unable to locate any suggestion, hint, motivation, or any teaching whatsoever that would lead one skilled in the art to apply his store queue technique to a multi-streaming microprocessor core that processes more than a single instruction thread. Furthermore, Meier does not suggest any of the afore-noted instruction queue limitations such as a read pointer, a write pointer, or retention of previously dispatched instructions.

For these reasons, Applicant respectfully requests that the rejections of claims 1, 3, and 5 be withdrawn.

With respect to claims 2, 10, 12, and 13, these claims depend from claim 1 and add further limitations that are neither anticipated nor made obvious by Dubey, Meier, or Dubey and Meier in combination. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 2, 10, 12, and 13.

With respect to claim 4, this claim depends from claim 3 and add further limitations that are neither anticipated nor made obvious by Dubey, Meier, or Dubey and Meier in combination. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejection of claim 4.

With respect to claims 22-23, these claims depend from claim 5 and add further limitations that are neither anticipated nor made obvious by Dubey, Meier, or Dubey and Meier in combination. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 22-23.

The Examiner rejected claims 7-9, 11, 14-17, and 18-21 under 35 U.S.C. 103(a) as being unpatentable over Dubey in view of Meier in further view of Levy et al., U.S. Patent Application Publication No. 20001/0004755 (hereinafter, Levy). More specifically, the

Examiner noted that the combination of Dubey and Meier does not teach eight instruction buffers, but that Levy teaches a processor executing a maximum of eight threads simultaneously. The Examiner thus concluded that it would have been obvious to one of ordinary skill in the art to have implemented 8 instruction buffers for 8 threads in the system of Dubey as suggested by Levy because Levy teaches that with 8 threads stalling drops and provides the greatest choice of instructions to issue.

Applicant respectfully traverses and notes that nowhere in any of the cited references, as argued above in disputation of the rejections of claims 1, 3, and 5, can one skilled obtain any sort of motivation or urging to provide instruction queues having a read pointer, a write pointer, or to provide for retention of previously dispatched instruction in a multi-streaming microprocessor core. And since claims 7-9, 11, 14-17, and 18-21 depend from either claims 1, 3, or 5, each of which recited the above-argued limitations, it is respectfully requested that the rejections of claims 7-9, 11, 14-17, and 18-21 be withdrawn.

The Examiner also rejected claims 9, 11, 16-17, and 20-21 as being unpatentable over Dubey in view of Meier and further in view of Levy. Applicant respectfully traverses and asserts again that neither Dubey, Meier, or Levy, alone or in combination, provide any teachings that would lead one skilled to provide instruction queues having a read pointer, a write pointer, or to retain previously dispatched instructions. Accordingly, Applicant requests the withdrawal of the rejections of claims 9, 11, 16-17, and 20-21.

CONCLUSIONS

In view of the arguments advanced above, Applicant respectfully submits that claims 1-23 are in condition for allowance. Reconsideration of the rejections is requested, and allowance of the claims is solicited.

Applicant earnestly requests that the Examiner contact the undersigned practitioner by telephone if the Examiner has any questions or suggestions concerning this amendment, the application, or allowance of any claims thereof.

EXPRESS MAIL LABEL NUMBER: **EO 004 399 478 US**
DATE OF DEPOSIT: **4/4/05**
I hereby certify that this paper is being deposited with the U.S. Postal Service Express Mail Post Office to Addressee Service under 37 C.F.R. §1.10 on the date shown above and is addressed to Mail Stop **PETITION**, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450.

Respectfully submitted,
HUFFMAN PATENT GROUP, LLC

By ______________________
RICHARD K. HUFFMAN
Reg. No. 41,082
Tel.: (719) 575-9998

Date 4\4\05